

Docket No.: 61282-074

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of : Customer Number: 20277
: :
Takenobu TANI : Confirmation Number:
: :
Serial No.: : Group Art Unit:
: :
Filed: April 16, 2004 : Examiner: Unknown
: :
For: MICROPROCESSOR EQUIPPED WITH POWER CONTROL FUNCTION, AND INSTRUCTION
CONVERTING APPARATUS

CLAIM OF PRIORITY AND
TRANSMITTAL OF CERTIFIED PRIORITY DOCUMENT

Mail Stop CPD
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

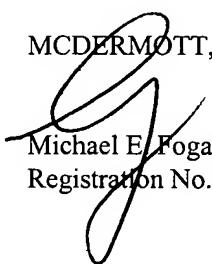
In accordance with the provisions of 35 U.S.C. 119, Applicant hereby claims the priority of:

Japanese Patent Application No. 2003-111666, filed April 16, 2003

cited in the Declaration of the present application. A certified copy is submitted herewith.

Respectfully submitted,

MCDERMOTT, WILL & EMERY


Michael E. Fogarty
Registration No. 36,139

600 13th Street, N.W.
Washington, DC 20005-3096
(202) 756-8000 MEF:tlb
Facsimile: (202) 756-8087
Date: April 16, 2004

61282-074
TANU
April 16, 2004

日本国特許庁
JAPAN PATENT OFFICE

McDermott, Will & Emery

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日 2003年 4月16日
Date of Application:

出願番号 特願2003-111666
Application Number:

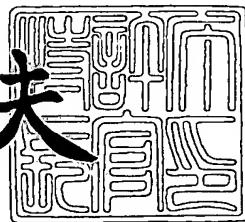
[ST. 10/C] : [JP2003-111666]

出願人 松下電器産業株式会社
Applicant(s):

2003年12月18日

特許庁長官
Commissioner,
Japan Patent Office

今井康夫





【書類名】 特許願
【整理番号】 5038240126
【提出日】 平成15年 4月16日
【あて先】 特許庁長官殿
【国際特許分類】 G06F 1/32
【発明者】
【住所又は居所】 大阪府門真市大字門真1006番地 松下電器産業株式会社内
【氏名】 谷 丈暢
【特許出願人】
【識別番号】 000005821
【氏名又は名称】 松下電器産業株式会社
【代理人】
【識別番号】 100105647
【弁理士】
【氏名又は名称】 小栗 昌平
【電話番号】 03-5561-3990
【選任した代理人】
【識別番号】 100105474
【弁理士】
【氏名又は名称】 本多 弘徳
【電話番号】 03-5561-3990
【選任した代理人】
【識別番号】 100108589
【弁理士】
【氏名又は名称】 市川 利光
【電話番号】 03-5561-3990

【選任した代理人】

【識別番号】 100115107

【弁理士】

【氏名又は名称】 高松 猛

【電話番号】 03-5561-3990

【選任した代理人】

【識別番号】 100090343

【弁理士】

【氏名又は名称】 栗宇 百合子

【電話番号】 03-5561-3990

【手数料の表示】

【予納台帳番号】 092740

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 0002926

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 電力制御機能を備えたマイクロプロセッサ及び命令変換装置

【特許請求の範囲】

【請求項 1】 電力制御機能を備えたマイクロプロセッサであって、
電力制御を行うか否かが予め設定される電力制御許可設定手段と、
前記マイクロプロセッサの動作資源に対して個別に電力制御を指定する電力制
御命令の実行時に、前記電力制御許可設定手段の設定に従って最終的に電力制御
信号の出力を制御する電力制御出力制御手段と、
を備えるマイクロプロセッサ。

【請求項 2】 請求項 1 記載のマイクロプロセッサであって、
前記電力制御命令が指定する電力制御を行うべき動作資源の情報を格納する電
力制御レジスタと、
前記電力制御レジスタに格納された情報に基づき前記動作資源のそれぞれに対
して電力制御を行わせる電力制御信号を出力する電力制御回路と、
前記電力制御許可手段の設定に従って前記電力制御信号をそれぞれゲートする
ゲート回路と、
を備えるマイクロプロセッサ。

【請求項 3】 電力制御機能を備えたマイクロプロセッサであって、
命令プログラムそれぞれに付与されるプログラム ID 毎に予め設定された電力
制御を行うか否かの情報を保持し、かつ実行中の命令プログラムのプログラム I
D が電力制御を行うように設定されたプログラム ID であるか否かを判定するプ
ロセッサ状態判定手段と、
前記マイクロプロセッサの動作資源に対して個別に電力制御を指定する電力制
御命令の実行時に、実行中の命令プログラムについての前記プロセッサ状態判定
手段の判定結果に従って最終的に電力制御信号の出力を制御する電力制御信号出
力制御手段と、
を備えるマイクロプロセッサ。

【請求項 4】 請求項 3 記載のマイクロプロセッサであって、
前記電力制御命令が指定する電力制御を行うべき動作資源の情報を格納する電

(A)

力制御レジスタと、

前記電力制御レジスタに格納された情報に基づき前記動作資源のそれぞれに対して電力制御を行わせる電力制御信号を出力する電力制御回路と、

前記プロセッサ状態判定手段の判定結果に従って前記電力制御信号をそれぞれゲートするゲート回路と、

を備えるマイクロプロセッサ。

【請求項5】 命令プログラムを所定のマイクロプロセッサの実行に適したように最適化する命令変換装置であって、

命令プログラム中に記述された指示文を参照して電力制御管理情報を抽出する電力制御管理手段と、

前記電力制御管理情報に基づき前記マイクロプロセッサの動作時に所定の長さの命令区間動作しない動作資源を検出する電力制御情報解析手段と、

前記電力制御情報解析手段の検出結果に基づき前記命令プログラムに電力制御に関する指示を付与する電力制御指示付与手段と、

を備える命令変換装置。

【請求項6】 請求項5記載の命令変換装置であって、

前記電力制御管理情報は前記所定の長さを指定する情報を含み、前記電力制御情報解析手段は前記電力制御管理情報に基づき前記所定の長さを切り換える命令変換装置。

【請求項7】 請求項5または6記載の命令変換装置であって、

命令毎にマイクロプロセッサの各動作資源が動作するか否かを示す情報を格納する命令別動作資源テーブル格納手段を備え、

前記電力制御情報解析手段は、前記命令別動作資源テーブル格納手段に格納された情報に基づいて、前記マイクロプロセッサの動作時に所定の長さの命令区間動作しない動作資源を検出する命令変換装置。

【請求項8】 命令プログラムを所定のマイクロプロセッサの実行に適したように最適化する命令変換装置であって、

前記マイクロプロセッサの動作時に所定の長さの命令区間動作しない動作資源を検出する電力制御情報解析手段と、

前記電力制御情報解析手段の検出結果に基づき前記命令プログラムに電力制御に関する指示を付与する電力制御指示付与手段とを備え、

前記電力制御情報解析手段は、動作資源の停止が可能な命令区間が長くなるよう命今プログラムを組み替える命今組み替え手段を備える命今変換装置。

【請求項 9】 請求項 8 記載の命今変換装置であって、

前記命今組み替え手段は、命今プログラム中の命今依存関係を維持しつつ命今を並べ替える命今並べ替え手段である命今変換装置。

【請求項 10】 請求項 8 記載の命今変換装置であって、

前記命今組み替え手段は、命今プログラム中の一の命今を前記一の命今と同一の処理結果を有する代替可能命今と差し替える命今差し替え手段である命今変換装置。

【請求項 11】 命今プログラムを所定のマイクロプロセッサの実行に適したように最適化する命今変換方法であって、

命今プログラム中に記述された指示文を参照して電力制御管理情報を抽出する電力制御管理ステップと、

前記電力制御管理情報に基づき前記マイクロプロセッサの動作時に所定の長さの命今区間動作しない動作資源を検出する電力制御情報解析ステップと、

前記電力制御情報解析ステップでの検出結果に基づき前記命今プログラムに電力制御に関する指示を付与する電力制御指示付与ステップと、
を備える命今変換方法。

【請求項 12】 請求項 11 記載の命今変換方法であって、

前記電力制御管理情報は前記所定の長さを指定する情報を含み、前記電力制御情報解析ステップは、前記電力制御管理情報に基づき前記所定の長さを切り換える命今変換方法。

【請求項 13】 請求項 11 または 12 記載の命今変換方法であって、

前記電力制御情報解析ステップは、命今毎にマイクロプロセッサの各動作資源が動作するか否かを示す情報を格納する命今別動作資源テーブルを参照して、前記マイクロプロセッサの動作時に所定の長さの命今区間動作しない動作資源を検出する命今変換方法。

【請求項 14】 命令プログラムを所定のマイクロプロセッサの実行に適したように最適化する命令変換方法であって、

前記マイクロプロセッサの動作時に所定の長さの命令区間動作しない動作資源を検出する電力制御情報解析ステップと、

前記電力制御情報解析ステップでの検出結果に基づき前記命令プログラムに電力制御に関する指示を付与する電力制御指示付与ステップとを備え、

前記電力制御情報解析ステップは、動作資源の停止が可能な命令区間が長くなるように命令プログラムを組み替える命令組み替えステップを含む命令変換方法。

【請求項 15】 請求項 14 記載の命令変換方法であって、

前記命令組み替えステップは、命令プログラム中の命令依存関係を維持しつつ命令を並べ替える命令変換方法。

【請求項 16】 請求項 14 記載の命令変換方法であって、

前記命令組み替えステップは、命令プログラム中の一の命令を前記一の命令と同一の処理結果を有する代替可能命令と差し替える命令変換方法。

【請求項 17】 請求項 5 から 10 のいずれか一項記載の命令変換装置で変換された命令プログラムを実行して内蔵する動作資源に対して個別に電力制御を行うマイクロプロセッサ。

【請求項 18】 請求項 11 から 16 のいずれか一項記載の命令変換方法で変換された命令プログラムを実行して内蔵する動作資源に対して個別に電力制御を行うマイクロプロセッサ。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、電力制御機能を備えたマイクロプロセッサ及び命令プログラムを所定のマイクロプロセッサの実行に適したように最適化する命令変換装置に関する。

【0002】

【従来の技術】

従来、マイクロプロセッサの消費電力を低減するために、ソフトウェア開発者が命令プログラムに電力制御命令を直接書き込むことが行われている。図21はソフトウェア駆動によりマイクロプロセッサの電力制御を行う命令プログラム例を示す図である。本例では区間2101においてデータメモリを使用していない。そこでソフトウェア開発者は2102の箇所にデータメモリを停止させる命令を挿入し、データメモリが再び使用される直前の2103の箇所にデータメモリの停止を解除する命令を挿入する。

【0003】

マイクロプロセッサは命令実行時に、デコーダで上記電力制御命令をデコードし、電力制御情報を抽出する。抽出された電力制御情報は電力制御レジスタに書き込まれ、電力制御回路がこの電力制御レジスタの状態に応じた電力制御を行う。このようにして、プログラムに挿入された上記2つの命令により、区間2101においてデータメモリを停止させることが可能である。

【0004】

本例ではデータメモリの例を示したが、マイクロプロセッサにおいて命令により電力制御可能な対象であれば、データレジスタ、データバス、演算器、周辺インターフェース、さらには一般の論理回路素子に及ぶマイクロプロセッサのあらゆる動作資源が制御可能である。

【0005】

しかしながら、上記の方法ではソフトウェア開発者が命令プログラム記述時に電力制御の開始や終了個所を判断し、直接命令プログラムに電力制御用命令の記述を行うため、ソフトウェア開発者の開発負担が大きい。また、指定個所が増加すると細かな制御指示を記述するのは困難である。さらに、電力制御の対象がメモリ構成や周辺インターフェースポート数等、マイクロプロセッサのシステム構成により異なる場合は、それぞれの機種に固有のプログラムが必要となるため、ソフトウェアの共用や流用が困難である。

【0006】

上記問題点を解決するために、命令プログラムを構成する命令の種類により各動作資源が動作するか否かに関する情報を予めテーブルに記録しておき、上記テ

ープルを用いて命令プログラムを解析し、解析結果に基づいて電力制御命令を挿入するコンパイラ等の命令変換装置が、特願 2003-17374 号明細書において提案されている。

【0007】

【発明が解決しようとする課題】

近年、ソフトウェア規模の増大とともにソフトウェアに不具合（バグ）が残り易くなり、ソフトウェアの安全性を保証することが困難になってきている。一方で、製品として出荷された後に不具合が発覚すると、製品の回収やソフトウェアの無償書き換えへと発展し、その被害は計り知れない。

【0008】

前記特願 2003-17374 号明細書に記載の技術は、コンパイラ等の命令変換装置で電力制御命令を自動挿入するものであるため、自動挿入による不具合が生ずる可能性も存在する。また、製品デバッグ時に不具合を発見したとしても、その不具合がソフトウェア開発者が記述した命令プログラムに起因するものか、あるいはコンパイラ等の命令変換装置が自動挿入した命令に起因するものかを判別することが困難である。

【0009】

しかしながら、例えばモバイル機器の機能の増大とともに低消費電力化への要求がさらに高まるなど、近年はあらゆる機器において低消費電力化への要求があり、ソフトウェア開発者が命令プログラムに手作業で電力制御命令を記述する方法では対応できなくなっている。そのため電力制御命令を自動挿入するコンパイラのような方法が必須となるが、電力制御命令の自動挿入により不具合が発生する可能性が大きな問題となる。

【0010】

また、電力制御に関するハードウェアは、他のハードウェアを停止させるものであるため、ハードウェアに関して初期不具合が発生する可能性が高い。このため、電力制御に関してはソフトウェア不具合とハードウェア不具合を判別することも重要になる。しかしながら、従来のソフトウェア開発者が命令プログラムに電力制御命令を直接書き込む方法では、プログラマが電力制御命令を含めてソフ

トウェアを作りこむので、これを判別することが難しい。このことはコンパイラによる電力制御命令の自動挿入による方法を用いても変わりはない。

【0011】

本発明は係る事情に鑑みてなされたものであり、命令プログラムを解析し電力制御命令を自動挿入するコンパイラ等による命令変換方法において、電力制御命令の自動挿入による不具合の発生を防止しソフトウェアの安全性を高めること、及び不具合の発生時にソフトウェア開発者が記述した命令プログラムに起因する不具合とコンパイラが自動挿入した命令に起因する不具合とを切り分けること、及び電力制御に関するハードウェアの不具合とソフトウェアによる不具合とを切り分けることを可能にする電力制御機能を備えたマイクロプロセッサ及び命令変換装置を提供することを目的とする。

【0012】

【課題を解決するための手段】

上記課題を解決するために、本発明の請求項1記載のマイクロプロセッサは、電力制御機能を備えたマイクロプロセッサであって、電力制御を行うか否かが予め設定される電力制御許可設定手段と、前記マイクロプロセッサの動作資源に対して個別に電力制御を指定する電力制御命令の実行時に、前記電力制御許可設定手段の設定に従って最終的に電力制御信号の出力を制御する電力制御出力制御手段とを備える。

【0013】

請求項2記載のマイクロプロセッサは、請求項1記載のマイクロプロセッサであって、前記電力制御命令が指定する電力制御を行うべき動作資源の情報を格納する電力制御レジスタと、前記電力制御レジスタに格納された情報に基づき前記動作資源のそれぞれに対して電力制御を行わせる電力制御信号を出力する電力制御回路と、前記電力制御許可手段の設定に従って前記電力制御信号をそれぞれゲートするゲート回路とを備える。

【0014】

請求項1または2記載のマイクロプロセッサによれば、電力制御許可手段により電力制御を行うか否かを決定することができるため、低消費電力よりもソフト

ウェアの安全性を優先したいときは電力制御を非許可として安全性を確保することができる。また、命令プログラムのデバッグ時には電力制御の許可／非許可を切り換えることにより、ソフトウェア開発者が記述した命令コードに起因した不具合と、命令変換によって挿入された電力制御命令に起因した不具合とを切り分けることが可能となる。

【0015】

請求項3記載のマイクロプロセッサは、電力制御機能を備えたマイクロプロセッサであって、命令プログラムそれぞれに付与されるプログラムID毎に予め設定された電力制御を行うか否かの情報を保持し、かつ実行中の命令プログラムのプログラムIDが電力制御を行うように設定されたプログラムIDであるか否かを判定するプロセッサ状態判定手段と、前記マイクロプロセッサの動作資源に対して個別に電力制御を指定する電力制御命令の実行時に、実行中の命令プログラムについての前記プロセッサ状態判定手段の判定結果に従って最終的に電力制御信号の出力を制御する電力制御信号出力制御手段とを備える。

【0016】

請求項4記載のマイクロプロセッサは、請求項3記載のマイクロプロセッサであって、前記電力制御命令が指定する電力制御を行うべき動作資源の情報を格納する電力制御レジスタと、前記電力制御レジスタに格納された情報に基づき前記動作資源のそれぞれに対して電力制御を行わせる電力制御信号を出力する電力制御回路と、前記プロセッサ状態判定手段の判定結果に従って前記電力制御信号をそれぞれゲートするゲート回路とを備える。

【0017】

請求項3または4記載のマイクロプロセッサによれば、実行中の命令プログラムが電力制御を行うように設定されているか否かに従って実際に電力制御を行うか否かを決定することができるため、例えば、信頼できるソフトウェア開発者が開発した命令プログラムでは電力制御を許可し、信頼できないソフトウェア開発者が開発した命令プログラムでは電力制御を非許可にすることができるので、命令プログラムの安全性を高めることができる。

【0018】

請求項5記載の命令変換装置は、命令プログラムを所定のマイクロプロセッサの実行に適したように最適化する命令変換装置であって、命令プログラム中に記述された指示文を参照して電力制御管理情報を抽出する電力制御管理手段と、前記電力制御管理情報に基づき前記マイクロプロセッサの動作時に所定の長さの命令区間動作しない動作資源を検出する電力制御情報解析手段と、前記電力制御情報解析手段の検出結果に基づき前記命令プログラムに電力制御に関する指示を与える電力制御指示付与手段とを備える。

【0019】

請求項6記載の命令変換装置は、請求項5記載の命令変換装置であって、前記電力制御管理情報は前記所定の長さを指定する情報を含み、前記電力制御情報解析手段は前記電力制御管理情報に基づき前記所定の長さを切り換えるものである。

【0020】

請求項7記載の命令変換装置は、請求項5または6記載の命令変換装置であって、命令毎にマイクロプロセッサの各動作資源が動作するか否かを示す情報を格納する命令別動作資源テーブル格納手段を備え、前記電力制御情報解析手段は、前記命令別動作資源テーブル格納手段に格納された情報に基づいて、前記マイクロプロセッサの動作時に所定の長さの命令区間動作しない動作資源を検出するものである。

【0021】

請求項5から7記載の命令変換装置によれば、ソフトウェア開発者は命令プログラム作成時に指示文を用いて電力制御を行う区間及び電力制御のレベルを指定することにより、電力制御を必要とする区間のみ任意のレベルの電力制御を行うことができるので、コードサイズの増大や動作速度の低下といった問題を最低限に抑え、きめ細やかな電力制御を行うことができる。また、電力制御命令の挿入や置換により安全性に問題が生じやすい個所は電力制御を行わないようにすることができるため、プログラムの安全性を高めることができる。

【0022】

請求項8記載の命令変換装置は、命令プログラムを所定のマイクロプロセッサ

の実行に適したように最適化する命令変換装置であって、前記マイクロプロセッサの動作時に所定の長さの命令区間動作しない動作資源を検出する電力制御情報解析手段と、前記電力制御情報解析手段の検出結果に基づき前記命令プログラムに電力制御に関する指示を付与する電力制御指示付与手段とを備え、前記電力制御情報解析手段は、動作資源の停止が可能な命令区間が長くなるように命令プログラムを組み替える命令組み替え手段を備える。

【0023】

請求項8記載の命令変換装置によれば、動作資源の停止が可能な命令区間が長くなるように命令プログラムを自動的に組み替えることができるため、電力制御情報解析手段が検出する動作資源が動作しない命令区間が長くなり、命令プログラム実行時の消費電力を低減することができる。

【0024】

請求項9記載の命令変換装置は、請求項8記載の命令変換装置であって、前記命令組み替え手段は、命令プログラム中の命令依存関係を維持しつつ命令を並べ替える命令並べ替え手段である。

【0025】

請求項9記載の命令変換装置によれば、命令プログラム中の命令の依存関係を維持したまま、動作資源の停止が可能な命令区間が長くなるように命令を並べ替えることができるため、電力制御情報解析手段が検出する動作資源が動作しない命令区間が長くなり、消費電力を効果的に低減することができる。

【0026】

請求項10記載の命令変換装置は、請求項8記載の命令変換装置であって、前記命令組み替え手段は、命令プログラム中の一の命令を前記一の命令と同一の処理結果を有する代替可能命令と差し替える命令差し替え手段である。

【0027】

請求項10記載の命令変換装置によれば、動作資源の停止が可能な命令区間が長くなるように命令プログラム中の命令を差し替え可能な命令へと差し替えることができるため、電力制御情報解析手段が検出する動作資源が動作しない命令区間が長くなり、消費電力を効果的に低減することができる。

【0028】

請求項11記載の命令変換方法は、命令プログラムを所定のマイクロプロセッサの実行に適したように最適化する命令変換方法であって、命令プログラム中に記述された指示文を参照して電力制御管理情報を抽出する電力制御管理ステップと、前記電力制御管理情報に基づき前記マイクロプロセッサの動作時に所定の長さの命令区間動作しない動作資源を検出する電力制御情報解析ステップと、前記電力制御情報解析ステップでの検出結果に基づき前記命令プログラムに電力制御に関する指示を付与する電力制御指示付与ステップとを備える。

【0029】

請求項12記載の命令変換方法は、請求項11記載の命令変換方法であって、前記電力制御管理情報は前記所定の長さを指定する情報を含み、前記電力制御情報解析ステップは、前記電力制御管理情報に基づき前記所定の長さを切り換えるものである。

【0030】

請求項13記載の命令変換方法は、請求項11または12記載の命令変換方法であって、前記電力制御情報解析ステップは、命令毎にマイクロプロセッサの各動作資源が動作するか否かを示す情報を格納する命令別動作資源テーブルを参照して、前記マイクロプロセッサの動作時に所定の長さの命令区間動作しない動作資源を検出する。

【0031】

請求項11から13に記載の命令変換方法によれば、ソフトウェア開発者は命令プログラム作成時に指示文を用いて電力制御を行う区間及び電力制御のレベルを指定することにより、電力制御を必要とする区間のみ任意のレベルの電力制御を行うことができるので、コードサイズの増大や動作速度の低下といった問題を最低限に抑え、きめ細やかな電力制御を行うことができる。また、電力制御命令の挿入や置換により安全性に問題が生じやすい個所は電力制御を行わないようになるため、プログラムの安全性を高めることができる。

【0032】

請求項14記載の命令変換方法は、命令プログラムを所定のマイクロプロセッ

サの実行に適したように最適化する命令変換方法であって、前記マイクロプロセッサの動作時に所定の長さの命令区間動作しない動作資源を検出する電力制御情報解析ステップと、前記電力制御情報解析ステップでの検出結果に基づき前記命令プログラムに電力制御に関する指示を付与する電力制御指示付与ステップとを備え、前記電力制御情報解析ステップは、動作資源の停止が可能な命令区間が長くなるように命令プログラムを組み替える命令組み替えステップを含む。

【0033】

請求項14記載の命令変換方法によれば、動作資源の停止が可能な命令区間が長くなるように命令プログラムを自動的に組み替えることができるため、電力制御情報解析ステップで検出する動作資源が動作しない命令区間が長くなり、命令プログラム実行時の消費電力を低減することができる。

【0034】

請求項15記載の命令変換方法は、請求項14記載の命令変換方法であって、前記命令組み替えステップは、命令プログラム中の命令依存関係を維持しつつ命令を並べ替えるものである。

【0035】

請求項15記載の命令変換方法によれば、命令プログラム中の命令の依存関係を維持したまま、動作資源の停止が可能な命令区間が長くなるように命令を並べ替えることができるため、電力制御情報解析ステップで検出される動作資源が動作しない命令区間が長くなり、消費電力を効果的に低減することができる。

【0036】

請求項16記載の命令変換方法は、請求項14記載の命令変換方法であって、前記命令組み替えステップは、命令プログラム中の一の命令を前記一の命令と同一の処理結果を有する代替可能命令と差し替えるものである。

【0037】

請求項16記載の命令変換方法によれば、動作資源の停止が可能な命令区間が長くなるように命令プログラム中の命令を差し替え可能な命令へと差し替えることができるため、電力制御情報解析ステップで検出される動作資源が動作しない命令区間が長くなり、消費電力を効果的に低減することができる。

【0038】

請求項17記載のマイクロプロセッサは、請求項5から10のいずれか一項記載の命令変換装置で変換された命令プログラムを実行して内蔵する動作資源に対して個別に電力制御を行うものである。

【0039】

請求項18記載のマイクロプロセッサは、請求項11から16のいずれか一項記載の命令変換方法で変換された命令プログラムを実行して内蔵する動作資源に対して個別に電力制御を行うものである。

【0040】

【発明の実施の形態】

以下、本発明の実施の形態について図面を参照しながら説明する。

(第1の実施形態)

図1は、本発明の第1の実施形態の電力制御機能を備えたマイクロプロセッサに係り、ユーザが記述した命令プログラムを実行オブジェクト形式の命令プログラムに変換するコンパイラ等の命令変換装置の構成と処理過程を示す図である。後述する命令翻訳手段を省略すれば、命令変換装置は実行オブジェクト形式の命令プログラムを変換するものであっても構わない。

【0041】

図1において、101はマイクロプロセッサで実行される命令プログラムである。命令プログラム101は、ニーモニックで記述されたアセンブラプログラムの命令列あるいはC言語やFORTRAN等のプログラム言語で記述された命令列のいずれの表現形式でも構わないが、以下の説明においては主にニーモニックで記述されたアセンブラプログラム例を用いて説明する。

【0042】

命令変換装置の電力制御情報解析手段は、命令毎動作資源解析手段102、命令別動作資源テーブル103、電力制御情報検出手段105からなる。命令別動作資源テーブル103には、命令毎にマイクロプロセッサ内のいずれの動作資源が動作するかの情報を予め記録しておく。

【0043】

命令毎動作資源解析手段102は、命令プログラム101を構成する命令毎に命令別動作資源テーブル103を参照し、いずれの動作資源が動作するかの情報を抽出し、命令毎動作資源使用データ104を出力する。電力制御情報検出手段105は、命令毎動作資源使用データ104の時系列的な解析を行い、いずれの個所にて動作資源の停止あるいは動作開始等の電力制御命令を挿入すると電力消費量を低減できるかを判断し、電力制御指示データ106を出力する。

【0044】

電力制御指示付与手段107は、電力制御指示データ106を基に、命令プログラム101に動作資源の停止あるいは動作開始等の命令の挿入を行い、電力最適化済命令プログラム108を生成する。命令翻訳手段109は電力最適化済命令プログラム108のプログラム言語を機械語へと翻訳し、マイクロプロセッサが実行すべき機械語命令コード列である電力最適化済実行オブジェクト110を生成出力する。命令プログラム101がすでに実行オブジェクト形式の場合は、命令翻訳手段109は省略され得る。

【0045】

図2は命令別動作資源テーブル103の構成例を示す図である。命令別動作資源テーブル103は、命令モードを示すフィールド201とマイクロプロセッサ内の動作資源を表すフィールド202の2つのフィールドを持つ二次元テーブルである。テーブル中の値は各命令モードの実行において各動作資源が動作するか否かを示している。命令別動作資源テーブルは必ずしも図2の形に限定されない。命令モードと動作資源の他にもパラメータが必要となった場合には三次元テーブルとなることもあり得る。

【0046】

本例では、テーブル中の”1”は動作資源が動作すること、“0”は動作資源が動作しないことを示している。例えば、レジスタ間の加算命令である「ADD R_x、R_y、R_z」の場合、マイクロプロセッサ動作時には「演算器A動作」、「データレジスタR0-R15」、「データレジスタR16-R31」のみが動作するので、テーブル中の対応する個所には”1”が記述されており、その他の動作資源は動作しないので、テーブル中の対応する個所には”0”が記述されている。

同様に、メモリの読み出し命令である「LD Ra、MEMy」では、テーブル中で”1”が記述されている「メモリRead動作」及び「ブロックA動作」のみが動作することになる。ここで、”0”と”1”の値は重要でなく、動作資源の動作と非動作を識別することが目的である。

【0047】

次に図3を用いて、命令プログラムが301の場合を例にとって、命令毎動作資源解析手段102の動作を説明する。命令毎動作解析手段102は命令プログラム301を入力として受け取ると、命令毎に命令別動作資源テーブル103の命令モードフィールドとの一致検索を行い、それぞれの命令に対応するマイクロプロセッサ内の動作資源を302のように出力する。302において、命令毎に”1”となっている動作資源が命令実行時に動作することを示している。命令毎動作資源解析手段102は、命令プログラムを構成する各命令について上記解析を行い、命令毎動作資源使用データ104を生成する。

【0048】

続いて図4を用いて、電力制御情報検出手段105の動作を説明する。電力制御情報検出手段105は、命令毎動作資源解析手段102が出力する命令毎動作資源使用データ104について、命令方向（時系列方向）に動作資源が動作するか否かについて解析を行う。解析方法の一例として、連続する”0”を検出する方法を説明する。図4に示す命令毎動作資源使用データにおいて、情報”0”はマイクロプロセッサがその命令実行において動作資源の動作を行わないことを示す。電力制御情報検出手段105は、命令毎動作資源使用データ104について、ある一定命令区間、例えば3命令区間以上連続する”0”を検出することにより、所定命令区間動作しない動作資源を検出できる。

【0049】

図4において「メモリRead動作」の解析を例にとり、3命令区間以上動作しない動作資源を検出する場合について説明する。「メモリRead動作」に対応する401に示したデータ列によると、SUB命令からSTR命令まで4命令の間は「メモリRead動作」を必要としていない。したがって電力制御情報検出手段105はこれを検出する。

【0050】

図4と同じ命令プログラムを例にとって、さらに電力制御指示付与手段107について説明を加える。図5は電力制御命令の挿入例である。図5に示すように、電力制御指示付与手段107は、電力制御情報検出手段105の出力する電力制御指示データ106に基づき、SUB命令の直前に「メモリRead動作」に関する回路停止を要求する命令コードを、STR命令の直後に回路動作開始の命令を挿入している、これにより、この区間において命令毎のデコードを必要とせずに「メモリRead動作」に関する回路停止を行うことができ、「メモリRead動作」に必要とされた消費電力を削減することができる。

【0051】

本例では3命令区間以上の動作資源の停止可能区間を検出しているが、電力制御情報検出手段105で検出される動作停止を行うことができる命令区間は、マイクロプロセッサで動作する各アプリケーションプログラムやシステムにより最適な電力制御のレベルを、1区間以上の長さで任意に設定することができる。

【0052】

また、プログラムのコンパイル時にオプションを用いてユーザが電力制御のレベルを指定する構成としても良い。例えば、オプション1を指定してコンパイルを行うと3区間以上動作しない動作資源に対して動作停止を行い、オプション2を指定してコンパイルを行うと5区間以上動作しない動作資源に対して動作停止を行うようにしても良い。電力制御命令を挿入しすぎると、命令増加によるプログラムのコードサイズの増大や動作速度の低下を招くことがある。このようにオプションを指定することにより、ソフトウェア開発者がプログラムのコードサイズと動作速度とを考慮しながら電力制御を行うことができる。なお、電力制御のレベルの指定対象としては区間の長さに限られるものではなく、様々に考えられる。例えば、電力制御の対象とする動作資源や、電力制御の方法（クロック周波数の低下、停止、電源供給のカット等）が考えられる。

【0053】

なお、図4の例ではSUB命令からMUL命令までの3命令間は「メモリWrite動作」も必要としていない。したがって本例では、この情報を検出するこ

とで、動作資源個別の電力制御のみを行う場合よりさらに上流側で電力制御を行うことができる。すなわち、命令プログラムに「メモリ Read 動作」や「メモリ Write 動作」に関する制御指示を付与するのみならず、メモリやメモリ制御回路自体を完全に停止させる制御指示を付与することもできる。

【0054】

また、動作資源の動作停止または開始と加算やビットシフト等の通常演算との両方を行う命令を1命令として命令仕様において定義している場合には、電力制御命令の挿入ではなく、命令の置換によって電力制御に関する情報を命令プログラムに付与するようにしても良い。

【0055】

以上説明した命令変換装置はコンピュータプログラムにより実現することができる。このプログラムはアセンブラーに組み込むことにより、ソフトウェア開発者が命令プログラム101をアセンブルすれば電力最適化済実行オブジェクト110までを作成するようになる。これにより、ソフトウェア開発者は何も意識する必要がなく、ただ命令プログラムを記述しアセンブルするだけで、マイクロプロセッサの消費電力は従来のものより低減される。もちろん、命令変換装置をアセンブラーに組み込まず、独立した電力制御用命令変換プログラムとすることもできる。

【0056】

また、ソフトウェア開発者の記述言語がC言語やFORTRANのようなプログラム言語である場合にはコンパイラに組み込むことも可能である。プログラム言語を用いることによりソフトウェア開発者はより簡単にプログラムを記述することができる。

【0057】

上記説明では、ソフトウェア開発者が記述した命令プログラムを解析し、電力最適化した後にプログラム言語で記述された命令プログラムから機械語で記述された命令プログラム（実行オブジェクト形式）への翻訳を行っているが、命令プログラムを実行オブジェクトに翻訳した後に、実行オブジェクトを解析して電力最適化済実行オブジェクトを作成しても何ら問題はない。さらに、このような命

令変換装置を回路で実現し、マイクロプロセッサに組み込むこともできる。

【0058】

命令変換装置により命令変換され作成された電力最適化済実行オブジェクト110をマイクロプロセッサは実行する。マイクロプロセッサは命令実行時に、命令デコーダにおいて電力最適化済実行オブジェクトに挿入された電力制御命令をデコードすることにより電力制御情報を抽出する。抽出された電力制御情報は電力制御レジスタに書き込まれ、電力制御回路がこの電力制御レジスタの状態に応じた電力制御を行う。

【0059】

図6は電力制御レジスタの構成例を示す図である。図6において、それぞれの電力制御対象は命令別動作資源テーブルの動作資源を表わすフィールドに対応し、例えば”メモリRead動作”、“メモリWrite動作”、“演算器A動作”、“演算器B動作”等に対応している。電力制御レジスタでビットが”1”となっている電力制御対象の動作資源が回路停止の対象となる。図6においては、電力制御対象2はビット”1”であるので回路停止の対象となるが、電力制御対象1、3、4は回路停止の対象とはならない。

【0060】

図7は本発明の第1の実施形態に係るマイクロプロセッサにおける電力制御機能の構成を示すブロック図である。図7において、命令デコーダ701は命令をデコードし各動作資源へ指示を与える。また、命令デコーダ701は電力制御命令をデコードしたときには電力制御情報を抽出する。電力制御レジスタ702は命令デコーダ701が抽出した電力制御情報を保持するレジスタであり、電力制御回路703は電力制御レジスタに書き込まれた電力制御情報に基づいて電力制御を行う回路である。

【0061】

さらに、電力制御許可レジスタ704はマイクロプロセッサが電力制御を行うかどうかを決定するレジスタであり、電力制御許可レジスタ704からの出力信号は電力制御回路が出力する電力制御信号とともにゲート回路群705への入力となる。ゲート回路群705の出力は各電力制御対象へと供給される。

【0062】

さらに具体的な動作を説明する。電力制御回路703は電力制御レジスタ702の値を参照し、回路停止の対象となる電力制御対象へと供給される電力制御信号を”H”にする。電力制御許可レジスタ704は電力制御が許可されているときは”H”を、電力制御が非許可のときは”L”を出力する。したがって電力制御が許可されているときは電力制御回路703から出力される電力制御信号がそのまま電力制御対象へと出力されるが、電力制御が非許可のときはゲート回路で抑止され、電力制御対象へは”L”しか供給されない。各電力制御対象は供給される電力制御信号が”H”的きに回路を停止する。

【0063】

電力制御許可レジスタ704を設けることにより、低消費電力よりも動作の安全性を優先したいときは、電力制御を非許可として安全性を確保することができる。また、命令プログラムのデバッグ時には電力制御の許可／非許可を切り換えることにより、ソフトウェア開発者が記述した命令コードに起因したソフトウェア的な不具合と、命令変換によって挿入された電力制御命令に起因した不具合とを切り分けることが可能となる。ここで命令変換によって挿入された電力制御命令に起因した不具合とは、電力制御に関するハードウェアに起因する不具合と挿入された電力制御命令に起因するソフトウェア的な不具合とを含む。

【0064】

さらに電力制御許可レジスタ704を設けることにより電力制御に関するハードウェアの不具合も切り分けることができる。具体的に説明すると、まずソフトウェア開発者がソフトウェアを開発して電力制御を非許可で実行する。このとき生じる不具合を①とする。この場合であってもソフトウェア開発者は最低限の電力制御命令を記述しているものとする。次に開発したソフトウェアを、電力制御を許可して実行する。このとき生じる不具合を②とする。すると、①と②の不具合の差分は電力制御に関するハードウェアに起因する不具合と言うことになる。さらに開発したソフトウェアを、電力制御命令を自動挿入する命令変換装置を用いて命令変換し電力制御を許可して実行する。このとき生じる不具合を③とする。すると、②と③の不具合の差分は命令変換によって挿入された電力制御命令に

よるソフトウェア的な不具合ということになる。このように命令変換、電力制御の許可／非許可を切替えながらデバッグを行うことにより種々の不具合を切り分けることができる

【0065】

(第2の実施形態)

図8は本発明の第2の実施形態に係るマイクロプロセッサにおける電力制御機能の構成を示すブロック図である。図8において、図7と同一の構成要素については同一の符号を付して説明を省略する。図8に示す電力制御機能においては、図7の電力制御許可レジスタ704に代えて、プロセッサ状態判定回路801を備えている。

【0066】

一般にマイクロプロセッサが実行する命令プログラムには部分毎にプログラムIDが付与されている。マイクロプロセッサは上記プログラムIDによって権限が設定される。本実施形態においては、プロセッサ状態判定回路801がプログラムIDを判断することにより、電力制御を行うか否かを決定する。電力制御を行うと判断したときにはゲート回路群705への出力信号を”H”にし、電力制御を行わないと判断したときにはゲート回路群705への出力を”L”にする。したがって、電力制御を行わないと判断したときはゲート回路は”L”しか出力しないため電力制御は行われない。

【0067】

図9はプロセッサ状態判定回路801が保持するID-電力制御テーブルの一例であり、プログラムIDと電力制御を許可するか否かの対応を示している。プロセッサ状態判定回路801はプログラムIDが入力されると、ID-電力制御テーブルのプログラムIDを一致検索することにより電力制御を行うかどうかを決定する。図9の例によると、プログラムIDがID1、ID2、ID4であるときはゲート回路群705への出力信号を”H”とし、電力制御回路が出力する電力制御信号をそのまま電力制御対象へと出力するが、プログラムIDがID3であるときはゲート回路群705への出力信号を”L”とするので電力制御回路703が出力する電力制御信号は抑止され、電力制御は行われない。

【0068】

上記のようにプログラムIDにより電力制御を行うか否かを決定することで、例えば、信頼できるソフトウェア開発者が開発した命令プログラムでは電力制御を許可し、信頼できないソフトウェア開発者が開発した命令プログラムでは電力制御を非許可にすることことができ、命令プログラムの実行における安全性を高めることができる。

【0069】

なお、プログラムIDはプログラム中の部分毎（例えばタスク毎）に設定される場合や、プログラム全体について設定される場合がありえるが、いずれの方法であれ、権限を設定できる方法であれば本発明に含まれるものである。

【0070】

（第3の実施形態）

本実施形態では、ソフトウェア開発者は命令プログラム作成時に指示文を用いて電力制御を行う区間及び電力制御のレベルを指定する。命令変換装置は上記指示文を参照し、電力制御情報検出手段を制御する。すなわち、指示文により電力制御を行うように指定された区間については、指定されたレベルに応じて電力制御可能な動作資源を検出し、指定されていない区間については電力制御可能な動作資源を検出しない。

【0071】

図10はソフトウェア開発者が記述する命令プログラムの一例を示す図である。本実施形態においては指示文の一例としてコメント分を用いた例で説明を行う。ソフトウェア開発者は電力制御情報をを行う区間の最初にコメント文”#pragma POWER_CONT_ON_Level1”を挿入して電力制御の開始を指示し、電力制御情報をを行う区間の最後に”#pragma POWER_CONT_OFF”を挿入して電力制御終了を指示している。上記2つのコメント文に挟まれた区間はLevel1で電力制御が行われる。

【0072】

図11は本実施形態における電力制御のレベルと実際に行われる電力制御との対応の一例を示す図である。この例では、Level0ではコードサイズの増大

などの問題が生じない置換可能な命令が定義されている動作資源のみを検出し、Level 1、2、3ではそれぞれ10、5、3区間以上動作しない動作資源を検出する。

【0073】

図12は、本発明の第3の実施形態の電力制御機能を備えた命令変換装置の構成と処理過程を示す図である。ここで、図1と同一の構成要素については同一符号を付し説明を省略する。図12においては、図1に示した命令変換装置の構成に加えて、電力制御情報管理手段1201を備えている。

【0074】

電力制御情報管理手段1201は命令プログラムから”#”から始まるコメント文を抽出する。抽出したコメント文が電力制御をONとするものであるときは、指定されたレベルで電力制御可能な動作資源を検出するように電力制御情報検出手段105に指示をする。抽出したコメント文が電力制御をOFFするものであるときには電力制御情報検出手段105の機能を停止する。

【0075】

本実施形態においては、電力制御を必要とする区間のみで任意のレベルの電力制御を行うことができるので、コードサイズの増大、動作速度の低下といった問題を最低限に抑え、きめ細やかな電力制御を行うことができる。

【0076】

さらに本実施形態においては、電力制御命令の挿入や置換を行うことにより安全性に問題が生じやすい個所は電力制御を行わないようになりますので、さらに命令プログラムの安全性を高めることができる。例えば、条件分岐が多用された個所などは電力制御命令の挿入や置換によりバグが生じやすい。このような個所は電力制御を行わないように制御することができる。

【0077】

なお、電力制御のレベルの指定対象は本実施形態に限られるものではなく、例えばLevel 0ではメモリに関する電力制御のみを行い、Level 1ではメモリと算術演算に関する演算器のみの制御を行い、Level 2で全ての動作資源について電力制御を行うといった動作資源を指定する方法や、Level 1に応じ

て電力制御の方法（クロック周波数の低下、停止、電源供給のカット等）を変化させる方法など様々な方法が考えられる。

【0078】

（第4の実施形態）

図13は、本発明の第4の実施形態の電力制御機能を備えた命令変換装置の構成と処理過程を示す図である。ここで、図1と同一の構成要素については同一符号を付し説明を省略する。図13においては、図1に示した命令変換装置の構成に加えて、命令並べ替え手段1301及びその出力である命令毎動作資源使用データ1302を備えている。

【0079】

マイクロプロセッサが有する動作資源には消費電力を考慮して予め優先度が設けられている。命令並べ替え手段1301においては、命令毎動作資源使用データ104を入力し、命令プログラムにおける命令間の依存関係を解析して並び替えが可能な命令を抽出する。そして、より長い区間で優先度の高い動作資源が停止できるように抽出された命令の順序を並び替え、その結果が命令毎動作資源使用データ1302として電力制御情報検出手段105へと出力される。

【0080】

図14は命令並べ替え前の命令プログラムの一例とその命令毎動作資源使用データ104を示している。本実施形態の説明においてはメモリ動作が最も優先度が高いものとする。図14の命令プログラムでは、”LD R1, Mem (A1)”と”LD R1, Mem (A1)”においてメモリRead動作を行うため、メモリWrite動作は7命令区間停止可能であるものの、メモリ動作としては最大で2命令区間しか停止させることができない。そこで本実施の形態においては、命令並べ替え手段1301において命令の依存関係の解析を行い、命令を並べ替えることによって停止可能な命令区間を長くし、より消費電力を低減する。

【0081】

図15は図14の命令プログラムにおける命令の依存関係を説明する図である。例えば、命令1502において用いられるレジスタR0の値は命令1501に

よって決定される。このため、命令1501と命令1502の順序を入れ替えると処理結果が違うものとなり、入れ替えることはできない。また、命令1503で用いられるレジスタR5の値は命令1502によって決定される。このため、命令1502と命令1503の順序も入れ替えることはできない。上記のような関係を依存関係といい、図中では矢印で結ばれる円と矩形の順序を入れ替えることができない。しかし、それ以外の命令は順序を入れ替えても正しく計算される。

【0082】

図16は命令並べ替え手段の構成をさらに詳しく示した図である。入力される命令毎動作資源使用データ104は、依存関係解析部1601によって上記のように命令間の依存関係が解析された後、依存関係に関する情報を付与され、依存関係解析データ1602として組み合わせ試行部1603へと出力される。組み合わせ試行部1603は依存関係を維持しながら命令プログラムの並べ替えを組み合わせ試行し、優先度の高い動作資源の停止可能区間が最も長くなるように命令の並べ替えを行う。

【0083】

図17は命令並べ替え後の命令プログラムとその命令毎動作資源使用データ1302を示している。本実施例では依存関係を崩さないようにして2つのLD命令を実行の順序が早くなるように並べ替えている。これによりメモリRead動作もメモリWrite動作と同じように5命令区間停止可能となるので、メモリ全体としての動作を5命令区間停止させることができる。

【0084】

このように命令並べ替え手段1301を設けることにより、優先度の高い、すなわち消費電力の大きな動作資源が停止する命令区間を長くするように命令の並べ替えを行うことができるので、さらなる低消費電力を実現することができる。

【0085】

なお、命令並べ替え手段1301の実施形態は上記の記載例に限られるものではなく、命令並べ替えによりさらなる低消費電力を実現するものであれば良い。例えば、動作資源毎に優先度を設け、優先度の高い動作資源をより長い命令区間

停止させる方法は簡単であるが、命令プログラムによっては命令並べ替えにより消費電力が低くなるとは限らない。そこで、動作資源が停止する命令区間の総和が長くなるように命令を並べ替える方法、あるいは動作資源ごとに想定される消費電力を予め定義しておき、マイクロプロセッサが実動作したときに消費電力がどれくらいになるかを監視しながら命令並べ替えを行う方法等、様々な他の実施形態が考えられる。

【0086】

本実施形態は、第3の実施形態と組み合わせることにより、命令プログラムの所定の個所のみで命令並べ替えを行うことも可能である。これによって命令並べ替えが有効な個所のみで命令並べ替えを行うと良い。特にループ部分のように高い頻度で実行される個所で命令並べ替えは有効である。

【0087】

(第5の実施形態)

図18は、本発明の第5の実施形態の電力制御機能を備えた命令変換装置の構成と処理過程を示す図である。ここで、図1と同一の構成要素については同一符号を付し説明を省略する。図18においては、図1に示した命令変換装置の構成に加えて、差し替え可能な命令の一覧を記憶した差し替え可能命令一覧テーブル1802と、命令差し替え手段1801及びその出力である命令毎動作資源使用データ1803を備えている。

【0088】

ここで差し替え可能な命令とは、命令を差し替えても同一の処理結果が得られるような命令のことをいう。マイクロプロセッサが有する動作資源には消費電力を考慮して予め優先度が設けられている。命令差し替え手段1801は差し替え可能命令一覧テーブル1802を参照し、命令の差し替えの結果で優先度の高い動作資源の停止可能な命令区間が長くなるようであれば実際に命令の差し替えを行う。

【0089】

図19は命令差し替え前の命令プログラムの一例とその命令毎動作資源使用データ104を示している。ここで、MUL R3, R0, 0x0002とはレジ

スタR0の値に2を乗算し、レジスタR3に格納する命令である。したがって、レジスタR0に格納された値を1ビット上位にシフトする命令（SFT R3, R0, 0x0001）と得られる処理結果は変わらないが、使う演算器が乗算器であるか、シフタであるかの点で異なっている。MUL R3, R0, 0x0004とSFT R3, R0, 0x0002も同様の関係である。なお、乗算器はシフタよりも優先度の高いものとする。ここで図19の命令プログラムではシフタは3命令区間停止させることが可能であるものの、乗算器は2命令区間しか停止させることができない。そこで本実施の形態においては命令差し替え手段1801において上記2つのMUL命令をSFT命令へと差し替える。

【0090】

図20は命令差し替え後の命令プログラムと命令毎動作資源使用データ1803である。MUL命令をSFT命令へと差し替えることにより、乗算器は6命令区間停止可能となっている。これによりマイクロプロセッサの消費電力はさらに低減される。

【0091】

なお、命令差し替え手段1801の実施形態は上記の記載例に限られるものではなく、命令差し替えによりさらなる低消費電力を実現するものであれば良い。例えば、動作資源毎に優先度を設け、優先度の高い動作資源をより長い命令区間停止させる方法は簡単であるが、命令プログラムによっては命令差し替えにより消費電力が低くなるとは限らない。そこで、動作資源が停止する命令区間の総和が長くなるように命令を差し替える方法、あるいは動作資源ごとに想定される消費電力を予め定義しておき、マイクロプロセッサが実動作したときに消費電力がどれくらいになるかを監視しながら命令差し替えを行う方法等、様々な他の実施形態が考えられる。

【0092】

本実施形態は、第3の実施形態と組み合わせることにより、命令プログラムの所定の個所のみで命令並べ替えを行うことも可能である。これによって命令並べ替えが有効な個所のみで命令並べ替えを行うと良い。特にループ部分のように高い頻度で実行される個所で命令並べ替えは有効である。

【0093】

また、本発明において動作資源を停止させる方法は、クロック供給の停止、クロック周波数の低下、供給電圧の制御、動作資源を構成するトランジスタの閾値電圧の制御など様々に考えられるが、動作資源を低消費電力状態にするものであれば特に方法は問わない。

【0094】

【発明の効果】

以上説明したように、本発明によれば、電力制御許可手段の設定、あるいは実行中の命令プログラムのプログラムIDにより電力制御を行うか否かを決定することができるため、命令変換装置によって電力制御命令を自動付与された命令プログラムを実行する場合でもソフトウェアの安全性を高めることができる。また、ソフトウェアの不具合をソフトウェア開発者が記述した命令プログラム自体に起因する不具合と、コンパイラで自動挿入された命令に起因する不具合とに切り分けることが可能となる。

【0095】

また本発明によれば、命令プログラム中の指示文を用いて電力制御を行う区間及び電力制御のレベルを自由に設定できるので、コードサイズの増大や動作速度の低下といった問題を最低限に抑え、きめ細やかな電力制御を行うことができる。また、電力制御命令の挿入や置換により安全性に問題が生じやすい個所は電力制御を行わないようにすることができるため、プログラムの安全性を高めることができる。

【0096】

また本発明によれば、命令の並べ替えや差し替えにより動作資源が停止可能な命令区間が長くなるように命令プログラムを自動的に組み替えることができるため、電力制御情報解析において検出する動作資源が動作しない命令区間が長くなり、消費電力を効果的に低減することができる。

【図面の簡単な説明】

【図1】

命令変換装置の基本構成と処理過程を示す図。

【図2】

命令別動作資源テーブルの構成例を示す図。

【図3】

命令毎動作資源解析手段による命令毎動作資源使用データの出力例を示す図。

【図4】

電力制御情報検出手段による電力制御指示データの出力例を説明する図。

【図5】

電力制御指示付与手段による電力制御命令の挿入例を示す図。

【図6】

電力制御レジスタの構成例を示す図。

【図7】

本発明の第1の実施形態に係るマイクロプロセッサの構成を示すブロック図。

【図8】

本発明の第2の実施形態に係るマイクロプロセッサの構成を示すブロック図。

【図9】

本発明の第2の実施形態に係るID-電力制御テーブルの一例を示す図。

【図10】

コメント文を用いて電力制御区間を指定した命令プログラムの一例を示す図。

【図11】

電力制御のレベルとその電力制御内容の対応例を示す図。

【図12】

本発明の第3の実施形態の命令変換装置の構成と処理過程を示す図。

【図13】

本発明の第4の実施形態の命令変換装置の構成と処理過程を示す図。

【図14】

命令並べ替え前の命令プログラム例と命令毎動作資源使用データを示す図。

【図15】

命令の依存関係を説明する図。

【図16】

命令並べ替え手段の構成を示す図。

【図17】

命令並べ替え後の命令プログラムと命令毎動作資源使用データを示す図。

【図18】

本発明の第5の実施形態の命令変換装置の構成と処理過程を示す図。

【図19】

命令差し替え前の命令プログラム例と命令毎動作資源使用データを示す図。

【図20】

命令差し替え後の命令プログラムと命令毎動作資源使用データを示す図。

【図21】

従来のマイクロプロセッサの電力制御を行う命令プログラム例を示す図。

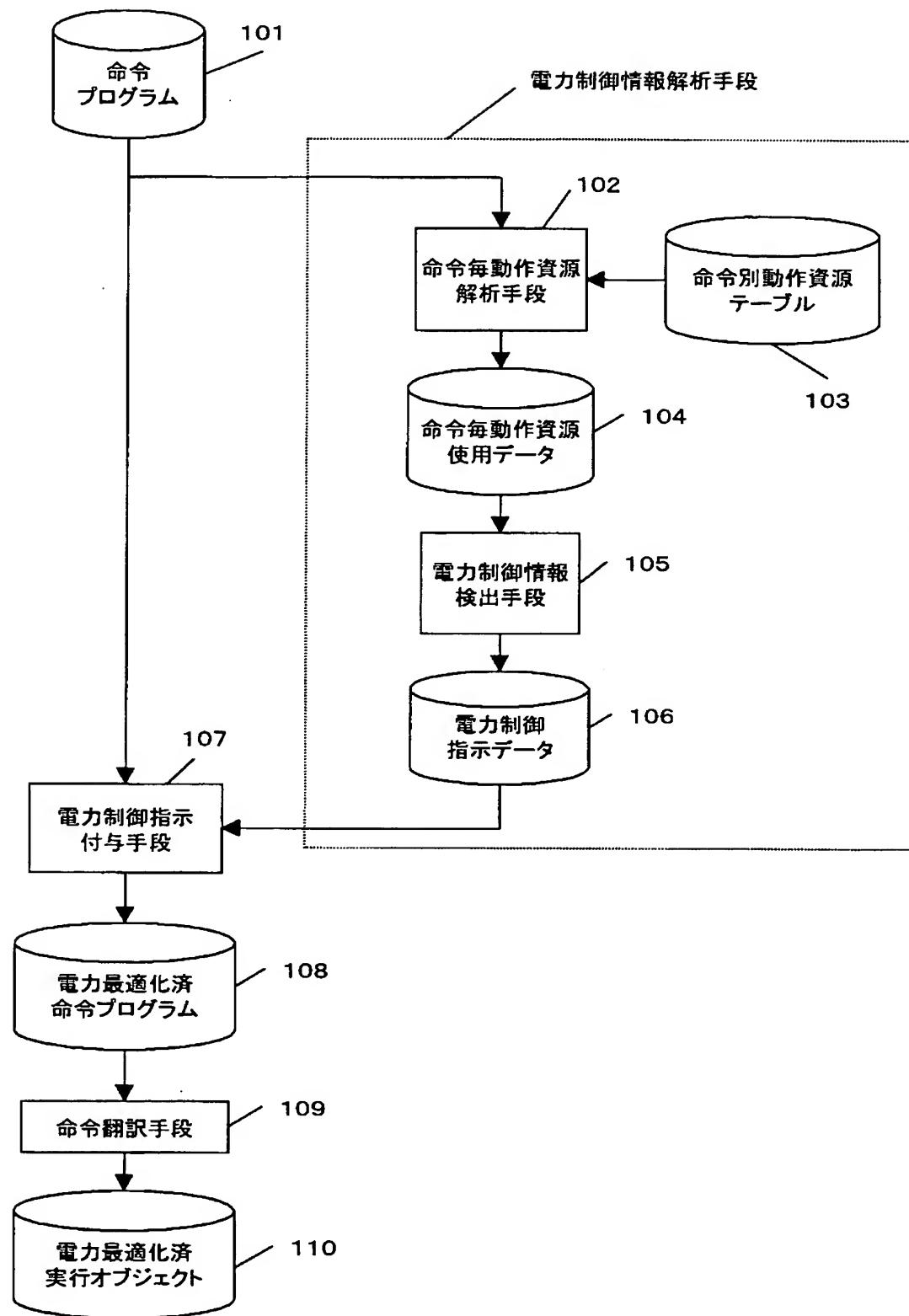
【符号の簡単な説明】

- 101 命令プログラム
- 102 命令毎動作資源解析手段
- 103 命令別動作資源テーブル
- 104 命令毎動作資源使用データ
- 105 電力制御情報検出手段
- 106 電力制御指示データ
- 107 電力制御指示付与手段
- 108 電力最適化済命令プログラム
- 109 命令翻訳手段
- 110 電力最適化済実行オブジェクト
- 201 命令モードフィールド
- 202 動作資源フィールド
- 301 命令プログラム例
- 302 命令毎動作資源使用データの出力例
- 401 電力制御指示データの出力例
- 701 命令デコーダ
- 702 電力制御レジスタ

- 703 電力制御回路
- 704 電力制御許可レジスタ
- 705 ゲート回路群
- 801 プロセッサ状態判定回路
- 1201 電力制御管理手段
- 1301 命令並べ替え手段
- 1302 命令並べ替え後の命令毎動作資源使用データ
- 1501～1503 命令プログラム中の命令
- 1601 依存関係解析部
- 1602 依存関係解析データ
- 1603 組み合わせ試行部
- 1801 命令差し替え手段
- 1802 差し替え可能命令一覧テーブル
- 1803 命令差し替え後の命令毎動作資源使用データ
- 2101 電力制御を行う区間
- 2102 データメモリを停止させる命令
- 2103 データメモリの停止を解除する命令

【書類名】 図面

【図 1】



【図2】

202 動作資源															
201 命令モード															
ADD Rx,Ry,Rz	メモリ Read 動作	メモリ Write 動作	演算器 A 動作	演算器 B 動作	演算器 C 動作	分歧ユニット A 動作	ブロック A 動作	ブロック B 動作	ブロック C 動作	周辺インターフェース動作	並列命令コードユニット	データレジスタ R0 - R15	データレジスタ R16 - R31	アドレス演算ユニット	即位ユニット
ADD Rx, MEMy, MEMz	0 0 1 0 0 0 0 0 0 0 0 0 1 1 0 0	1 0 1 0 0 0 0 0 0 0 0 0 1 1 1 0	0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 1	1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0	1 0 1 0 0 0 1 0 0 0 0 1 1 0 0 0	0 1 0 0 0 0 1 1 0 0 0 0 1 0 0 0	0 0 0 1 0 0 0 0 0 0 0 0 1 1 0 0	0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1	0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0	0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0	0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0	0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 1	0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0	0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 1	
ADD Rx, Ry, imm	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
LD Ra, MEMy															
LD Ra, MEMy::ADD Rx, Ry, Rz															
STR Ra, MEMy															
MUL Rx, Ry, Rz															
IO ADDR															
JUMP Rx															
LOOP N															
:															

命令別動作資源テーブル

【図3】

The diagram illustrates the mapping of memory addresses 301 and 302 to resource usage matrices. Address 301 is associated with the first column of the matrix, and address 302 is associated with the second column.

	メモリ Read 動作	メモリ Write 動作	演算器 A 動作	演算器 B 動作	分岐ユニット A 動作	ブロック A 動作	ブロック B 動作	ブロック C 動作	周辺インターフェースA動作	並列命令デコードユニット	データレジスタR0-R15	データレジスタR16-R31	アドレス演算ユニット	即値ユニット
301	LD R1,Mem(A0)	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	LD R1,Mem(A1)	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	SUB R0,R1,0xffff	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	ADD R2,R5,R7	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	MUL R3,R0,R2	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	STR Mem(A4),R3	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	LD R4,Mem(A5)	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
302														

命令毎動作資源使用データ

【図4】

The diagram illustrates the mapping of memory address 401 to a detailed resource usage matrix. Address 401 is associated with the first column of the matrix.

	メモリ Read 動作	メモリ Write 動作	演算器 A 動作	演算器 B 動作	分岐ユニット A 動作	ブロック A 動作	ブロック B 動作	ブロック C 動作	周辺インターフェースA動作	並列命令デコードユニット	データレジスタR0-R15	データレジスタR16-R31	アドレス演算ユニット	即値ユニット
401	LD R1,Mem(A0)	1 0 0 0 0 0 1 0 0 0 0 1 0 1 0 0	LD R1,Mem(A1)	1 0 0 0 0 0 1 0 0 0 0 0 1 0 1 0	SUB R0,R1,0xffff	1 0 0 1 0 0 0 0 0 0 0 0 1 0 0 1	ADD R2,R5,R7	1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0	MUL R3,R0,R2	0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0	STR Mem(A4),R3	0 1 0 0 0 0 1 1 0 0 0 0 1 0 1 0	LD R4,Mem(A5)	1 0 0 0 0 0 1 0 0 0 0 0 1 0 1 0

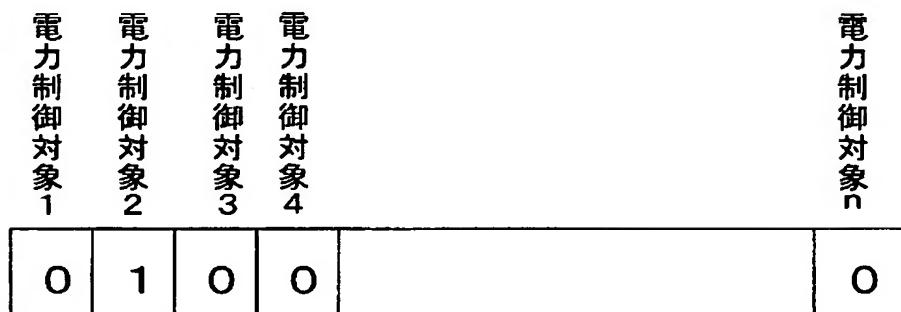
【図 5】

```

LD R1,Mem(A0)
LD R1,Mem(A1)
[SET PCR #Memory Read Stop]
SUB R0,R1,0xffff
ADD R2,R5,R7
MUL R3,R0,R2
STR Mem(A4),R3
[CLR PCR #Memory Read Stop]
LD R4,Mem(A5)

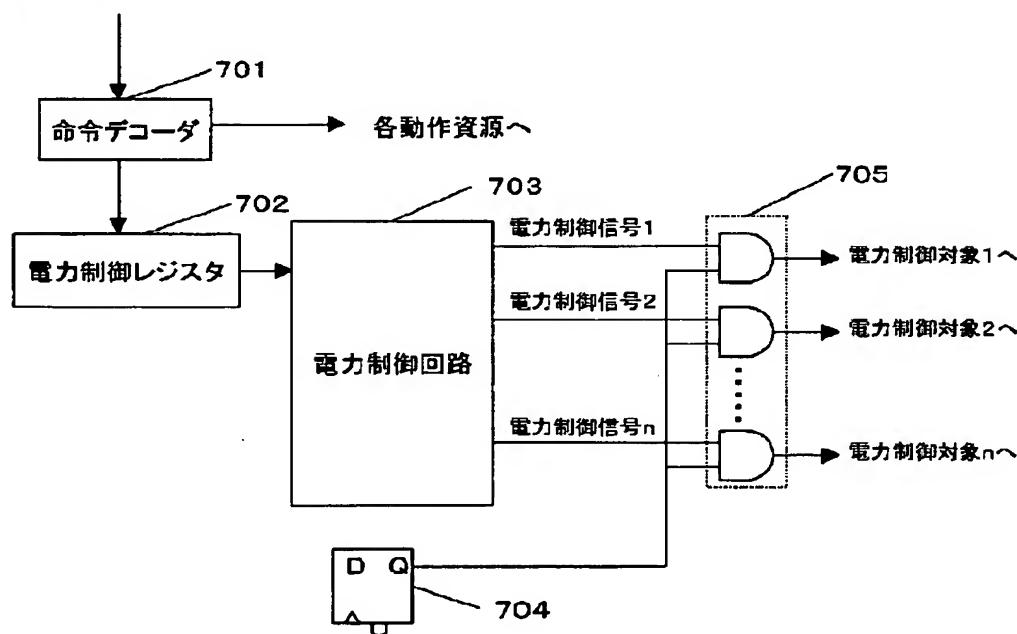
```

【図 6】



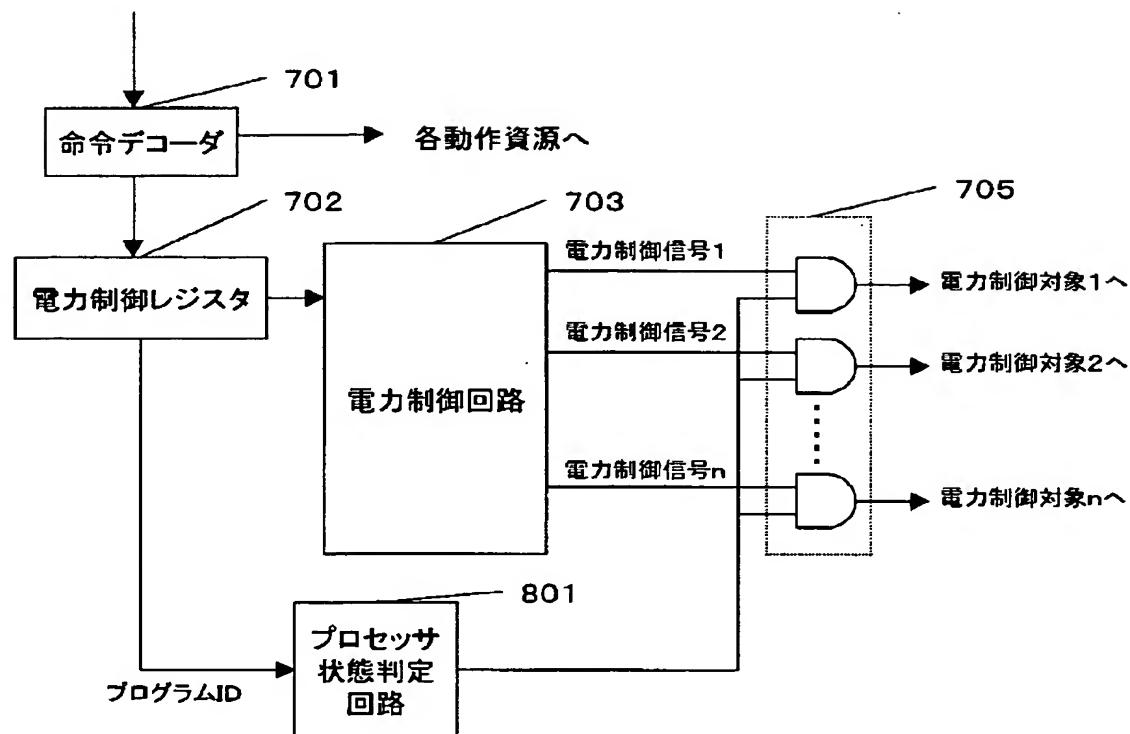
【図 7】

電力最適化済命令プログラム



【図8】

電力最適化済命令プログラム



【図9】

プログラムID	電力制御機能ON／OFF
ID1	ON
ID2	ON
ID3	OFF
ID4	ON

【図10】

```

data_a = data_b*1.75;
data_c = func_calc_d( in1, in2, in3 );

if ( cond_k == 1 ){ adrs1 = adrs1 + 8; }

#pragma POWER_CONT_ON_Level1
for ( i=0 ; i<256 ; i++ ){
    out_sum = out_sum * data_c[adrs1]
}
#pragma POWER_CONT_OFF

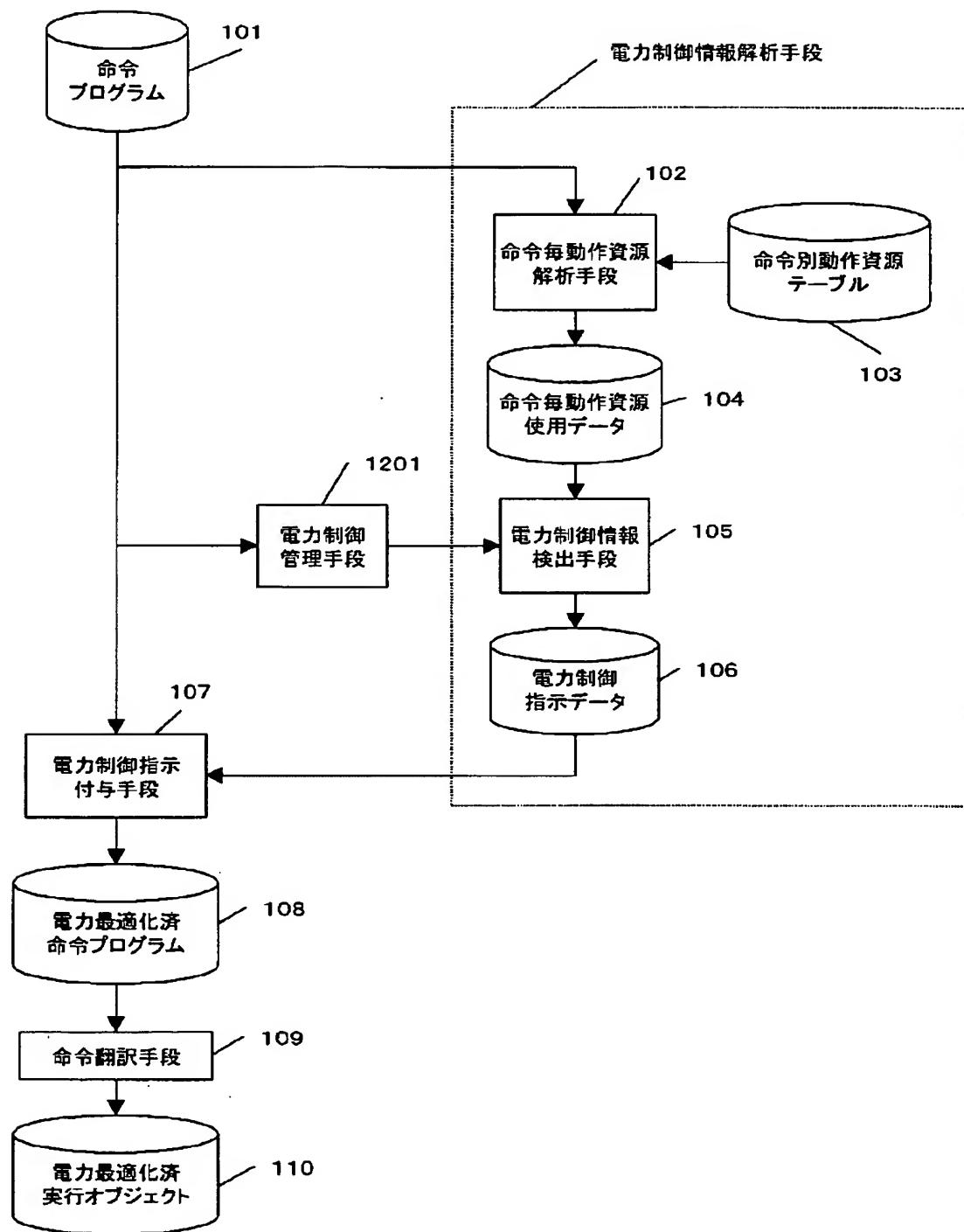
if ( out_sum > 24 ){ adrs1 = adrs1 + 32; }

```

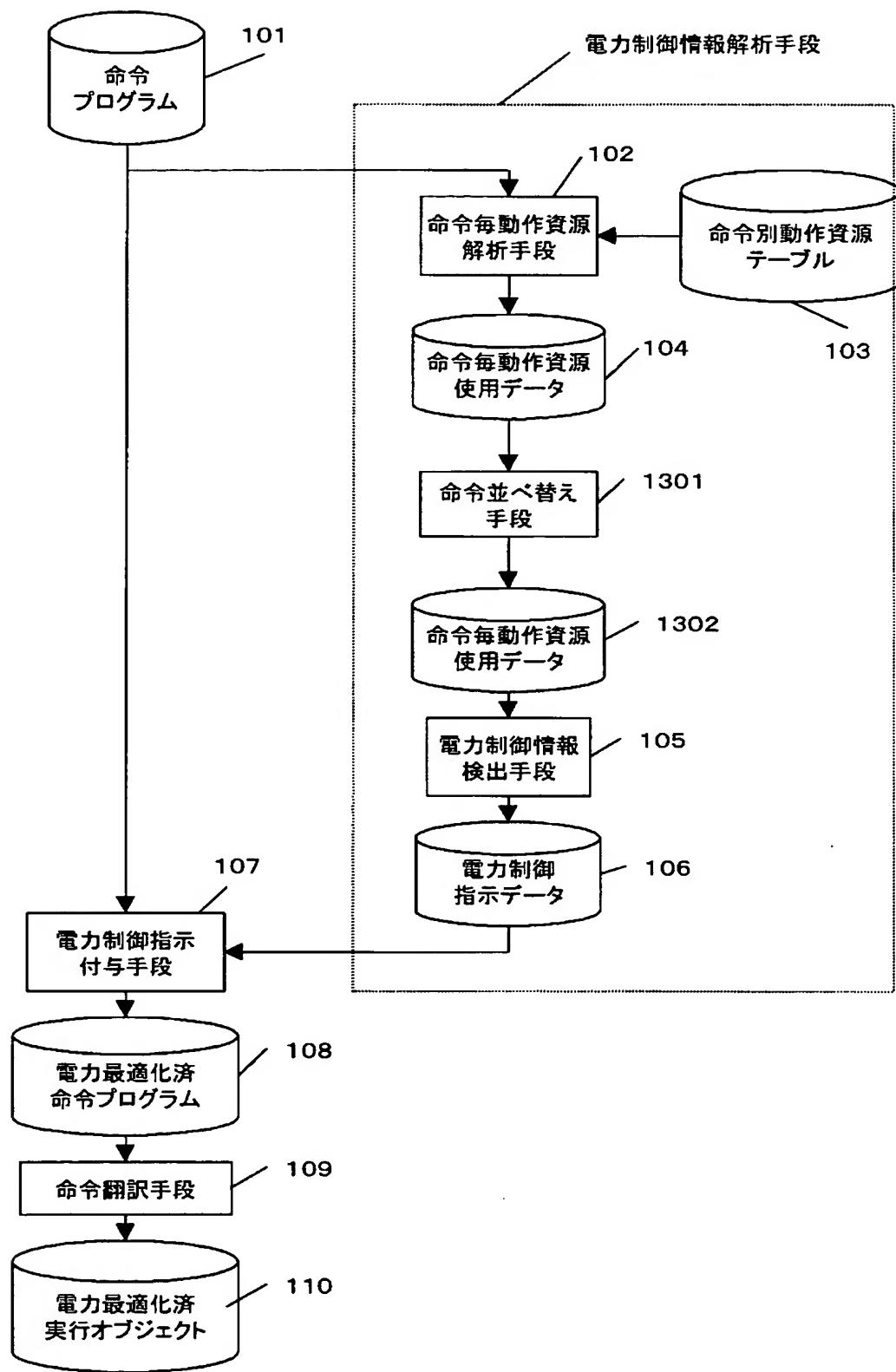
【図11】

レベル	制御内容
Level 0	命令置換によって停止可能な動作資源のみ検出
Level 1	10区間以上動作しない動作資源を検出
Level 2	5区間以上動作しない動作資源を検出
Level 3	3区間以上動作しない動作資源を検出

【図12】



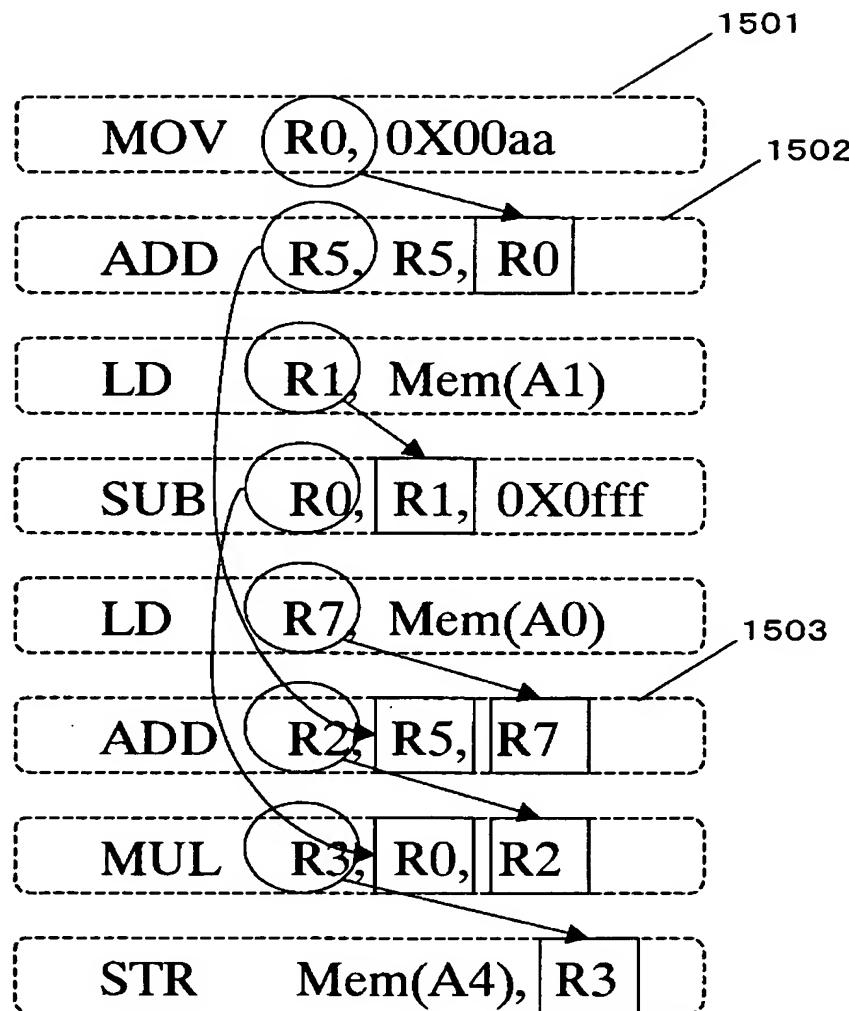
【図13】



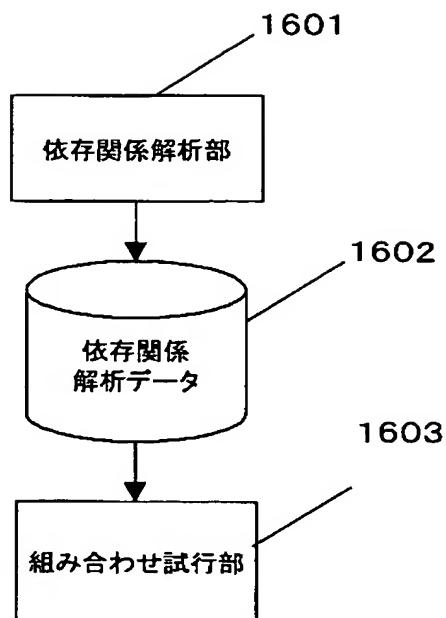
【図14】

	メモリRead動作	メモリWrite動作	演算器A動作	演算器B動作	分岐ユニットA動作	ブロックA動作	ブロックB動作	ブロックC動作	周辺インターフェース動作	並列命令コードユニット	データレジスタR0-R15	データレジスタR16-R31	アドレス演算ユニット	即値ユニット
MOV R0, 0X00aa	0	0	1	0	0	0	0	0	0	0	1	0	0	1
ADD R5, R5, R0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
LD R1, Mem(A1)	1	0	0	0	0	1	0	0	0	0	1	0	1	0
SUB R0, R1, 0X0fff	0	0	1	0	0	0	0	0	0	0	1	0	0	1
LD R7, Mem(A0)	1	0	0	0	0	1	0	0	0	0	1	0	1	0
ADD R2, R5, R7	0	0	1	0	0	0	0	0	0	0	0	0	0	0
MUL R3, R0, R2	0	0	0	1	0	0	0	0	0	0	1	0	0	0
STR Mem(A4), R3	0	1	0	0	0	1	1	0	0	0	1	0	1	0

【図15】



【図16】

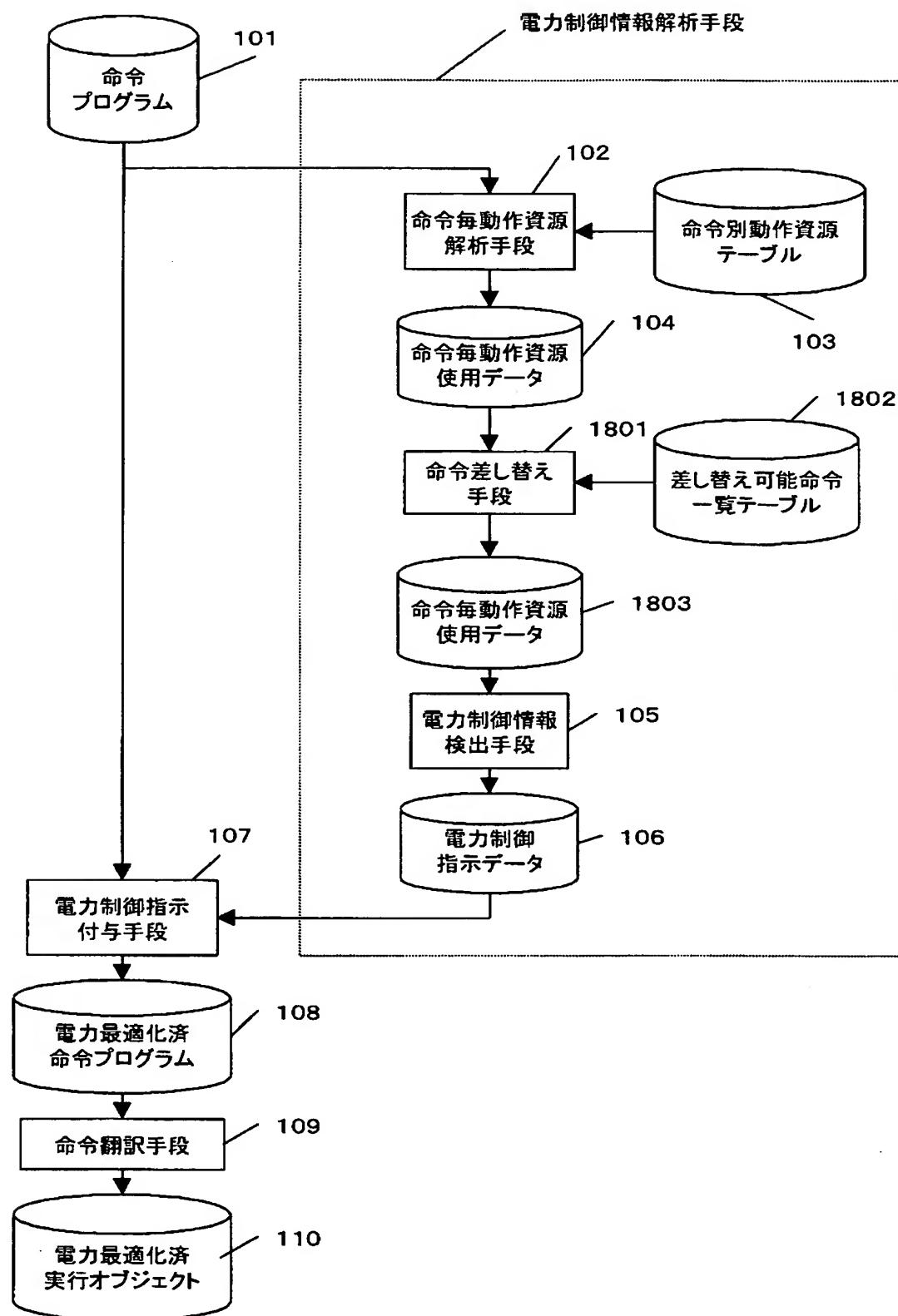


【図17】

メモリRead動作
 メモリWrite動作
 演算器A動作
 演算器B動作
 分岐ユニットA動作
 ブロックA動作
 ブロックB動作
 ブロックC動作
 周辺インターフェース動作
 並列命令コードユニット
 データレジスタR0-R15
 データレジスタR16-R31
 アドレス演算ユニット
 即値ユニット

	1	0	0	0	0	1	0	0	0	0	1	0	1	1	0
LD R1, Mem(A1)	1	0	0	0	0	1	0	0	0	0	1	0	1	0	0
LD R7, Mem(A0)	1	0	0	0	0	1	0	0	0	0	1	0	1	0	0
MOV R0, 0X00aa	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0
ADD R5, R5, R0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
SUB R0, R1, 0X0fff	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0
ADD R2, R5, R7	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
MUL R3, R0, R2	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0
STR Mem(A4), R3	0	1	0	0	0	1	1	0	0	0	1	0	1	0	0

【図18】



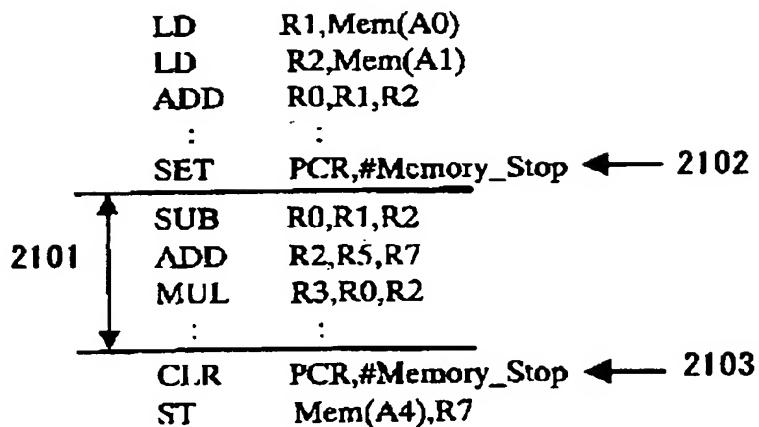
【図19】

	メモリRead動作	メモリWrite動作	シフター動作	乗算器動作	分岐ユニットA動作	ロックA動作	ロックB動作	ロックC動作	周辺インターフェース動作	並列命令コードユニット	データレジスターR0-R15	データレジスターR16-R31	アドレス演算ユニット	即値ユニット
STR Mem(A2), R9	0	0	1	0	0	0	0	0	0	0	1	0	0	1
MUL R3, R0, 0x0002	0	0	0	1	0	0	0	0	0	0	1	0	0	0
LD R1, Mem(A1)	1	0	0	0	0	1	0	0	0	0	1	0	1	0
MUL R5, R7, 0x0004	0	0	0	1	0	0	0	0	0	0	1	0	0	0
SUB R0, R1, 0xffff	0	0	1	0	0	0	0	0	0	0	1	0	0	0
ADD R2, R5, R7	0	0	1	0	0	0	0	0	0	0	0	0	0	0
MUL R3, R0, R2	0	0	0	1	0	0	0	0	0	0	1	0	0	0
STR Mem(A4), R3	0	1	0	0	0	1	1	0	0	0	1	0	1	0

【図20】

	メモリRead動作	メモリWrite動作	シフター動作	乗算器動作	分岐ユニットA動作	ロックA動作	ロックB動作	ロックC動作	周辺インターフェース動作	並列命令コードユニット	データレジスターR0-R15	データレジスターR16-R31	アドレス演算ユニット	即値ユニット
STR Mem(A2), R9	0	0	1	0	0	0	0	0	0	0	1	0	0	1
<u>SFT R3, R0, 0x0001</u>	0	0	1	0	0	0	0	0	0	0	1	0	0	0
LD R1, Mem(A1)	1	0	0	0	0	1	0	0	0	0	1	0	1	0
<u>SFT R5, R7, 0x0002</u>	0	0	1	0	0	0	0	0	0	0	1	0	0	0
SUB R0, R1, 0xffff	0	0	1	0	0	0	0	0	0	0	1	0	0	0
ADD R2, R5, R7	0	0	1	0	0	0	0	0	0	0	0	0	0	0
MUL R3, R0, R2	0	0	0	1	0	0	0	0	0	0	1	0	0	0
STR Mem(A4), R3	0	1	0	0	0	1	1	0	0	0	1	0	1	0

【図21】



【書類名】 要約書

【要約】

【課題】 マイクロプロセッサの動作資源に対して個別に電力制御を指定する電力制御命令が自動挿入された命令プログラムを実行するマイクロプロセッサにおいて、電力制御命令の自動挿入による不具合の発生を防止し、また不具合の発生時に命令プログラムの記述に起因する不具合とコンパイラが自動挿入した命令に起因する不具合とを切り分ける。

【解決手段】 電力制御命令が指定する電力制御を行うべき動作資源の情報を格納する電力制御レジスタ702と、電力制御レジスタ702に格納された情報に基づき動作資源のそれぞれに対して電力制御を行わせる電力制御信号を出力する電力制御回路703と、電力制御を行うか否かが予め設定される電力制御許可レジスタ704と、電力制御許可レジスタ704の設定により前記電力制御信号をそれぞれゲートするゲート回路705とをマイクロプロセッサに備える。

【選択図】 図7

特願2003-111666

出願人履歴情報

識別番号 [000005821]

1. 変更年月日 1990年 8月28日

[変更理由] 新規登録

住所 大阪府門真市大字門真1006番地
氏名 松下電器産業株式会社